

Eigenes Histogramm

Eigenes Histogramm

Es soll ein eigenes Histogramm erzeugt werden. Der Dateiname für das Skript ist `myhistogram.R`.

Funktion `myhistogram`

Programmieren Sie in R die Funktion `myhistogram`, die als Parameter `x` einen Vektor aus Zahlen erhält. Die Zahlen werden in `n` Intervalle einsortiert, und es wird gezählt, wie oft eine Zahl in einem Intervall vorkommt. Der Rückgabewert ist eine Liste mit den Einträgen `borders`, die die $n + 1$ Intervallgrenzen enthalten und `counts`, der die Anzahlen enthält.

- Die n Intervalle sollen gleich groß sein (Δb), d.h. für die Intervallgrenzen b_1, b_2, \dots, b_{n+1} gilt $\frac{b_{i+1} - b_i}{n} = \Delta b$ für $i = 1, 2, \dots, n$.
- Die äußeren Grenzen b_1 und b_n sollen als optionale Parameter `min` und `max` an die Funktion übergeben werden. Werte aus `x`, die zu keinem Intervall gehören, sollen ignoriert werden. Es wird aber eine Warnung ausgegeben, die sagt, welche Zahlen außerhalb des Bereichs liegen.
- Eine Zahl z gehört zum i -ten Intervall, falls $b_i \leq z < b_{i+1}$ gilt.

Bis auf `x` sollen alle Parameter optional sein. Überlegen Sie sinnvolle Default-Werte.

Es ist natürlich **nicht** erlaubt, in der eigenen Funktion andere Funktionen zu nutzen, die ein Histogramm erzeugen.

Hier ein Beispiel:

```
source("myhistogram.R")
x = seq(-5, 6, by = 1 / 3)

l = myhistogram(x, 5, -3)

print(l)
print(l$borders)
print(l$counts)
```

Beispieldaten

Hier zunächst zwei Beispiele.

Beispiel 1

Es wird eine Warnung ausgegeben:

```
## Warning in myhistogram(x, n = 10, min = -5, max = 6): Zahl(en) außerhalb
## Intervallgrenzen: 6
```

```
# Anmerkung: Bei VSCode auf macOS war es für 'Knit' nötig,
# in jedem Code-Chunk die source erneut anzugeben...
```

```
source("myhistogram.R")
x = seq(-5, 6, by = 1 / 3)
myhistogram(x, n = 10, min = -5, max = 6)
```

```
## Warning in myhistogram(x, n = 10, min = -5, max = 6): Zahl(en) außerhalb
## Intervallgrenzen: 6

## $borders
## [1] -5.0 -3.9 -2.8 -1.7 -0.6  0.5  1.6  2.7  3.8  4.9  6.0
##
## $counts
## [1] 4 3 3 4 3 3 4 3 3 3

solution1 = list(
  borders = c(-5.0, -3.9, -2.8, -1.7, -0.6, 0.5, 1.6, 2.7, 3.8, 4.9, 6.0),
  counts = c(4, 3, 3, 4, 3, 3, 4, 3, 3, 3)
)
expected_warning = 'Zahl(en) außerhalb Intervallgrenzen: 6'
```

Beispiel 2

```
source("myhistogram.R")
x = seq(-5, 6, by = 1 / 3)
myhistogram(x, n = 5, min = -10, max = 10)

## $borders
## [1] -10 -6 -2  2  6 10
##
## $counts
## [1]  0  9 12 12  1

solution2 = list(borders = c(-10, -6, -2, 2, 6, 10),
  counts = c(0, 9, 12, 12, 1))
```

Beispiel 3

Testen Sie nun hier Ihre Funktion mit weiteren Datensätzen.

```
source("myhistogram.R")
x3 = c(10, 12, 15, 20, 25, 30, 35, 40, 45)
myhistogram(x3, n = 5, min = 5, max = 30)

## Warning in myhistogram(x3, n = 5, min = 5, max = 30): Zahl(en) außerhalb
## Intervallgrenzen: 30, 35, 40, 45

## $borders
## [1]  5 10 15 20 25 30
##
## $counts
## [1] 0 2 1 1 1

solution3 = list(borders = c(5, 14, 23, 32, 41, 50),
  counts = c(2, 2, 2, 2, 1))
expected_warning = 'Zahl(en) außerhalb Intervallgrenzen: 30, 35, 40, 45'
```

Beispiel 4

Testen Sie nun hier Ihre Funktion mit weiteren Datensätzen.

```
source("myhistogram.R")
x2 = c(-2, -1, 0, 1, 2, 3, 4, 5)
myhistogram(x2, n = 4, min = -3, max = 6)
```

```
## $borders
## [1] -3.00 -0.75  1.50  3.75  6.00
##
## $counts
## [1] 2 2 2 2

solution4 = list(borders = c(-3, -0.75, 1.50, 3.75, 6.00),
                 counts = c(2, 2, 2, 2))
```

Beispiel 5

Testen Sie nun hier Ihre Funktion mit weiteren Datensätzen.

```
source("myhistogram.R")
x5 = rep(0, 10)
myhistogram(x5, n = 4, min = -3, max = 6)

## $borders
## [1] -3.00 -0.75  1.50  3.75  6.00
##
## $counts
## [1]  0 10  0  0

solution5 = list(borders = c(-3, -1.5, 0, 1.5, 3, 4.5, 6),
                 counts = c(0, 10, 0, 0, 0, 0))
```

Barplot

Nutzen Sie Ihre Funktion `myhistogram` und erzeugen Sie einen Barplot mit *ggplot*. Die *x*-Achse zeigt dabei die Mitte des Intervalls und die *y*-Achse die Anzahl der Elemente in dieser Klasse.

Tipp: Der Parameter `stat` von `geom_bar` ist wichtig.

Vervollständigen Sie den Chunk. Die Kommentare sollen zu Anweisungen umgewandelt werden:

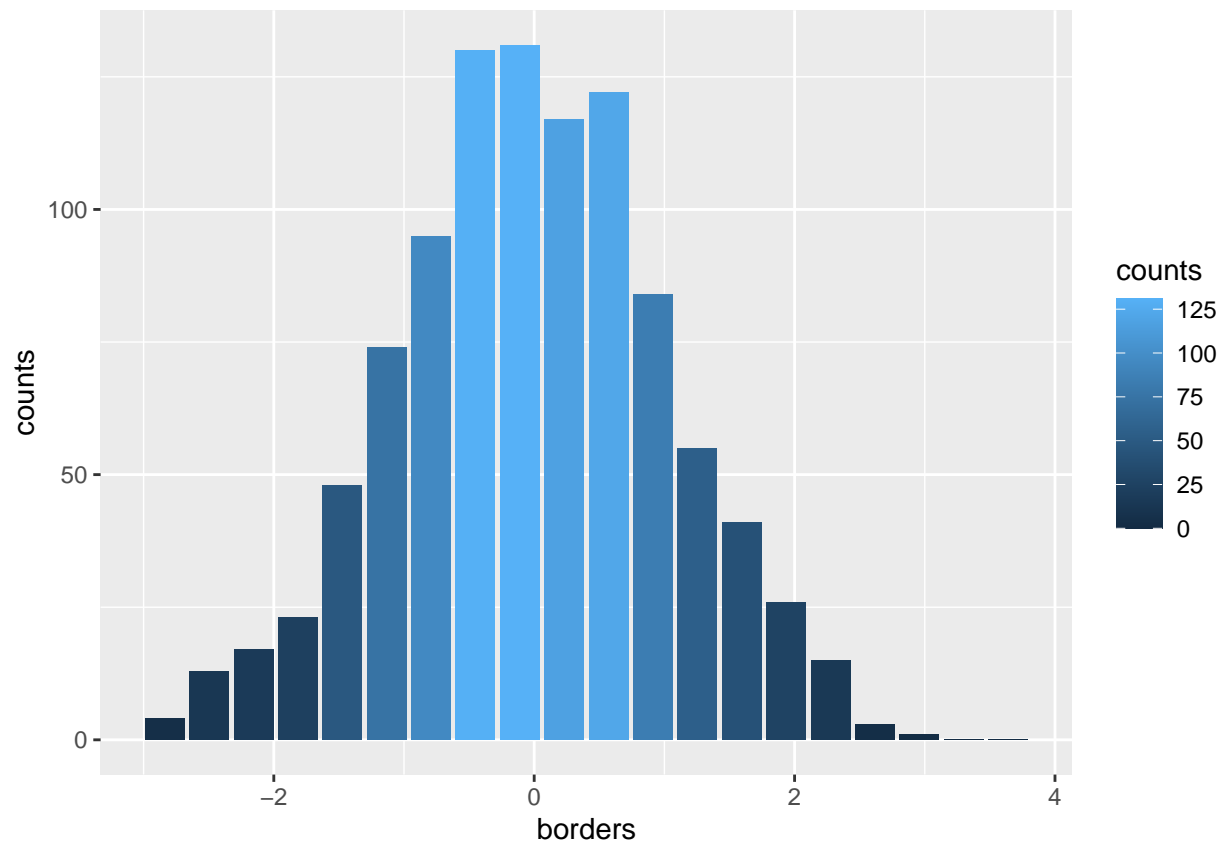
```
source("myhistogram.R")
library("ggplot2")
library("knitr")
set.seed(1)
x = rnorm(0, 1, n = 1000)
h = myhistogram(x, n = 20)

## Warning in myhistogram(x, n = 20): Zahl(en) außerhalb Intervallgrenzen:
## 3.81027668071067

mittelpunkte = (h$borders[-1] + h$borders[-length(h$borders)]) / 2

df = data.frame(borders = mittelpunkte, counts = h$counts[])

ggplot(df) +
  geom_bar(aes(x = borders, y = counts, fill = counts), stat = "identity")
```



```
ggsave(filename = 'myhistogram.jpg')
```

```
## Saving 6.5 x 4.5 in image
```